

(12)

EUROPEAN PATENT APPLICATION

(43) Date of publication:
 03.11.2004 Bulletin 2004/45

(51) Int Cl.7: G07F 17/32

(21) Application number: 04251400.0

(22) Date of filing: 11.03.2004

<div>(84) Designated Contracting States:</div> <div>AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IT LI LU MC NL PL PT RO SE SI SK TR</div> <div>Designated Extension States:</div> <div>AL HR LT LV MK</div>	<div>(71) Applicant: Rok Corporation Ltd</div> <div>Albrighton, Wolverhampton, WV7 3AU,</div> <div>West Midlands (GB)</div>
<div>(30) Priority: 28.04.2003 GB 0309623</div> <div>30.06.2003 GB 0315289</div>	<div>(72) Inventor: Bowall, Graham</div> <div>Surrey, CR0 7QJ (GB)</div>
	<div>(74) Representative: Geary, Stuart Lloyd et al</div> <div>Venner Shipley LLP</div> <div>20 Little Britain</div> <div>London EC1A 7DH (GB)</div>

(54)

Gaming system with remote user interface

(57)

A gaming system comprises a server (2) and a client (1). In response to a request from the client (1), the server (2) generates one or more random values to the client (1). The client (1) the generates an image giv-

ing the impression of the real-time generation of a received random value in response to a user input. The generation of the image is reported back to the server (1).

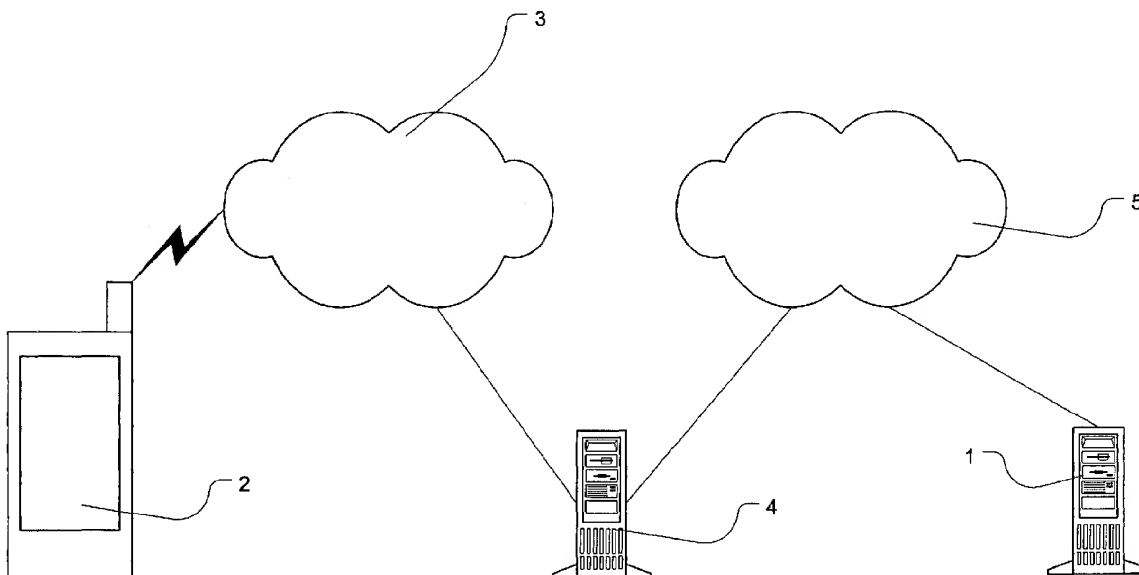


Figure 1

Description

[0001] The present invention relates to method providing a remote user interface in a gaming system, the method comprising sending a play request message from a client to a remote server, responding to the play request message at the server by generating a random play outcome value and transmitting the random play outcome value to the client and receiving said play outcome value at the client. The present invention also related to gaming system comprising a client including display means and a server remote from the client, wherein the client is configured for sending a play request message to the remote server, receive a play outcome value, provided by the server in response to said play request message and the server is configured for responding to the play request message by generating a random play outcome value and transmitting the random play outcome value to the client, and to a server system for a gaming system having a remote user interface.

[0002] Various form of gaming apparatus are well-known with the slot machine or "one-arm bandit" being a typical example. The common features of gaming apparatus are means for generating a random value, means for accepting a stake, a user input device for triggering the generation of the random value and means for displaying the random value. The random value may be constrained so that an apparatus will pay out a large percentage of the stake money received. Historically, slot machines displayed the random value by stopping reels, bearing indicia on their circumferential faces, so that a set of indicia is presented to the user. Cathode ray tubes have now largely replaced the old reels. However, the user is often presented with an image of spinning reels, imitating a electro-mechanical slot machine, on the cathode ray tube.

[0003] It is known for a client to send a request to a server and have a value returned in a gaming context. For example, a simple CGI (Common Gateway Interface) program emulating dice is disclosed at <http://www.irony.com/igroll.html>. This system suffers from the disadvantage that the dice rolling can only be performed when the end user can send requests to the server.

[0004] A solution to this problem is to provide the end user with the means to generate the outcome locally. This, however, has the problem that the outcome may need to be reported back, e.g. so that a prize can be claimed, and this reporting step makes this approach vulnerable to fraud.

[0005] An aim of the present invention is to provide a gaming system, and component parts thereof, which enables gaming using a client and a remote server which is less susceptible to fraud. A reduction in the susceptibility to fraud has been achieved from the insight that an end user need only be provided with the experience of game play verisimilarly. Thus, the game outcome can be determined at the server and communicated to the client, where the game play experience can subse-

quently be given to the user. Consequently, the grating of prizes or league positions is not dependent on easily spoofed reporting messages.

[0006] A method providing a remote user interface in a gaming system, according to the present invention is characterised by receiving a game start user input at the client after said reception of the play outcome value and generating an image at the client in dependence on said play outcome value in response to reception of said game start input.

[0007] A gaming system, according to the present invention is characterised by the client being configured to respond to a user input, after receiving the play outcome value, to generate an image at the client in dependence on the received play outcome.

[0008] A server system for a gaming system having a remote user interface, according to the present invention, is characterised by web server means making a client program available for download and dynamic content generating means, wherein the client program is configured for sending a play request message from a client to the web server means and generating an image at the client in dependence on a play outcome value, received from the web server means in response to a play request message, and in response to a user input made after reception of said play outcome value, and the dynamic content generating means is configured for responding to a play request message by generating a play outcome value and transmitting the random value to the requesting client.

[0009] The play outcome value is preferably random to some degree. In many gaming markets, controlled profile methodologies are implemented and the game software limits the volatility of the games to keep the game closer to a desired percentage payout as possible, i.e. random with less volatility.

[0010] It is to be understood that a substantial time may elapse between a play outcome value being received and the generation of the image. Thus, there is preferably no communication connection or session in operation between the client and the server when the image is being generated.

[0011] Preferably, the image is a moving image, e.g. the reels of a slot machine, racing horses or cards turning over, and said play outcome value determines the final state of said image.

[0012] Preferably, a plurality of play outcome values are generated and transmitted to the client in response to one play request message therefrom.

[0013] Preferably, the client reports the generation of the image to the server. Where a plurality play definition numbers are downloaded together, the generation of images is preferably reported when all of the corresponding images have been generated.

[0014] Preferably, communication between the client and the server uses http.

[0015] Preferably, the client is a mobile phone, a PDA or a communicator.

[0016] An embodiment of the present invention will now be described, by way of example, with reference to the accompanying drawings, in which:-

Figure 1 shows the components of a system embodying the present invention;

Figure 2 is a block diagram of the origin server of Figure 1;

Figure 3 is a first view of the user interface of the system of Figure 1;

Figure 4 is a second view of the user interface of the system of Figure 1;

Figure 5 is a flowchart illustrating the processing of a "buy" request by the origin server in Figure 1;

Figure 6 is a flowchart illustrating the game playing process at the client in Figure 1;

Figure 7 is a third view of the user interface of the system of Figure 1; and

Figure 8 is a fourth view of the user interface of the system of Figure 1.

[0017] Referring to Figure 1, a system embodying the present invention comprises an origin server 1, a client 2 comprising a mobile phone which supports J2ME (Java 2 Micro Edition), a mobile phone network 3, a WAP gateway 4 and the Internet 5.

[0018] Referring to Figure 2, the origin server 1 comprises an Apache web server 6 with a Tomcat servlet container 7, a database 8 and static content. The database 8 comprises user information, including user account data. The static content includes a gaming MIDlet jar file 9 and the corresponding descriptor jad file 10. Dynamic content is provided using JSP (Java Server Pages) and the servlet container 7 which hosts first and second servlets 11a, 11b.

[0019] In order to use the system shown in Figures 1 and 2, a user must register with the system and open an account with the operator of the origin server 1. In order to play games, the user must transfer funds to his/her account with the operator of the origin server 1. The details of the user, e.g. name and address, and the user's account are stored in the database in a conventional manner. On registering, the user is provided with username and password for accessing resources on the origin server 1.

[0020] In order to play a game, the user must first download the corresponding MIDlet 9 from the origin server 1. The MIDlet 9 may be downloaded before registration but cannot be used until after registration. "Over the air" provisioning of MIDlets in this way is conventional. The downloaded MIDlet 9 provides a user interface for the game. In the present example, the user interface simulates the reels of a slot machine. However, other graphics such as racing horses could be used.

[0021] Referring to Figure 3, having downloaded the necessary MIDlet 9, the user runs the MIDlet 9 immediately or at some later time and is presented with a screen comprising a message 12, giving the number of plays

outstanding, a menu 13 with the options "Play" 13a and "Buy" 13b and "OK" and "EXIT" command options 14a, 14b. If the user selects "Buy" 13b and the "OK" command option, the user is presented with a username and password entry screen 15 (Figure 4) by the MIDlet 9 which provides "OK" and "Back" command options 15a, 15b in addition to username and password textfields 15c, 15d. Selecting the "Back" option 15b takes the user back to the first screen shown in Figure 3. However, if the user selects the "OK" option 15a, the MIDlet 9 sends an http request to the origin server 1 and displays a screen with a message asking the user to wait while the request is processed. The request includes the entered username and password and a URL encoded string identifying the type of plays, being requested, as slot machine plays and the number being bought.

[0022] Referring to Figure 5, the request is received by the origin server 1 and the Apache web server 6 authenticates the user on the basis of the username and password in the request (step s1). If the user is not successfully authenticated, a 404 "access denied" response is sent back to the client 2 (step s2). Alternatively, a custom XML-formatted error message may be returned. The MIDlet 9 handles 404 or custom error responses by displaying the username and password entry screen 15 again but this time with a message informing the user of the authentication failure.

[0023] If the user is successfully authenticated (step s1), the request is passed to the first servlet 11a. The first servlet 11a checks whether the user has sufficient funds in his/her account to pay for the plays by accessing the database 8 (step s3) and, if not, sends an "insufficient funds" message to the client 1 (step s4). The client 1 will respond to such a message by displaying a message inviting the user to transfer further funds to his/her account with the operator of the origin server 1.

[0024] If there are sufficient funds to pay for the new plays (step s3), the first servlet 11a calculates twenty random values (steps s5 and s6). These values are calculated in much the same way as in a conventional slot machine. The set of twenty values is given a unique ID and this ID is stored in the database with the total monetary value of the wins, if any, among the twenty random values (step s7).

[0025] A response message, containing a string representation of the ID and the twenty random values, is then sent to the client 1 (step s8) and the user's account is debited by the stake for twenty plays (step s9).

[0026] The client 1 receives the response and stores the ID and random values, and then displays a message reporting the successful purchase of twenty new plays.

[0027] Referring to Figure 6, if the user selects the "Play" option 13a (Figure 3), the MIDlet 9 reads the next random value from its store (step s101). Referring to Figure 7 also, the MIDlet 9 then generates an moving image 21 representing the spinning reels of an electro-mechanical slot machine (step s102). The "reels" are made to appear to stop in turn. The symbol (bell, bar,

cherries, plum etc.) displayed on each "reel" when it stops is determined by the current random value.

[0028] Referring to Figure 8, when all of the "reels" have come to rest, a suitable commiseratory or congratulatory message 22 is added to the displayed image together with an invitation to play again 23 or return to the main screen 24 (step s103).

[0029] At the end of the play, the MIDlet 9 determines whether the play was the last of purchased set of twenty plays (step s104). If the play was the last of a set, the MIDlet 9 sends a request message to the origin server 1 which includes the ID of the completed play set and the user's username and password (step s105).

Assuming that the user is authenticated successfully, the request is passed to the second servlet 11b. The second servlet 11b retrieves the win amount for the play set, identified by the ID, from the database 8 and transfers any win amount to the user's account. The second servlet 11b then sends an acknowledge response to the client 1.

[0030] If the user selects the play again option (step s106), flow returns to step s101, otherwise the main screen (Figure 3) is displayed.

[0031] The request sent at step s105, may include the symbols displayed at the end of each play and the associated win values for validation at the origin server 2.

[0032] A second embodiment of the present invention will now be described.

[0033] In the second embodiment, the user registration and downloading of the MIDlet are as described above.

[0034] Referring to Figure 9, having downloaded the necessary MIDlet 9, the user runs the MIDlet 9 immediately or at some later time and is presented with a screen comprising a message 112.

[0035] If, when running the MIDlet 9, the user has not yet downloaded any plays or has used all of his/her downloaded plays remaining the user is presented with "Buy" and "Exit" options 114, 115. In order to play the game, the user selects the "Buy" option 114. If the user selects the "Buy" option 114, the user is presented with a username and password entry screen 15 (Figure 4) by the MIDlet 9 which provides "OK" and "Back" command options 15a, 15b in addition to username and password textfields 15c, 15d. Selecting the "Back" option 15b takes the user back to the first screen shown in Figure 3. However, if the user selects the "OK" option 15a, the MIDlet establishes a socket connection to the origin server 1 and sends a SOAP (Simple Object Access Protocol) message to the server requesting a packet of plays. The request message includes the users username and password and an id for the type of plays being requested, which in this example are slot machine plays. While the communication with the origin server 1 is taking place, the MIDlet 9 displays a screen with a message asking the user to wait while the request is processed.

[0036] Referring to Figure 10, the request is received

by the origin server 1 and the first servlet 11a authenticates the user on the basis of the username and password in the request (step s201). If the user is not successfully authenticated, an XML error response is sent back to the client 2 (step s202). Next, the first servlet 11a determines whether the user has used all previously downloaded plays as a security measure (step s203). If the database records show that the user has unused plays, a error response is sent (step s204). The MIDlet 9 handles the error response by displaying the username and password entry screen 15 again but this time with a message informing the user of the authentication failure or a screen displaying an appropriate error text.

[0037] If the user is successfully authenticated and has not plays left (steps s201 and s203), the first servlet 11a checks whether the user has sufficient funds in his/her account to pay for the plays by accessing the database 8 (step s204) and, if not, sends an "insufficient funds" message to the client 1 (step s4). The client 1 will respond to such a message by displaying a message inviting the user to transfer further funds to his/her account with the operator of the origin server 1.

[0038] If there are sufficient funds to pay for the new plays (step s204), the first servlet 11a calculates twenty random values (steps s206 and s207). These values are calculated in much the same way as in a conventional slot machine. The set of twenty values is given a unique ID and this ID is stored in the database with the total monetary value of the wins, if any, among the twenty random values (step s208).

[0039] A response message, containing a string representation of the ID and the twenty random values, i.e. the final reel positions for the twenty plays, and the win values of each play is then sent to the client 1 (step s209) and the user's account is debited by the stake for twenty plays (step s210).

[0040] The client 1 receives the response and stores the ID and random values, and then displays a screen showing a message 112 giving the number of plays left, a "Play" option 116 and an "Exit" option 115 (Figure 11). The screen shown in Figure 11 is also presented when the user runs the MIDlet 9 with plays in hand.

[0041] If the user selects the "Exit" option from the screen shown in Figure 4, the MIDlet 9 terminates.

[0042] Referring again to Figure 6, if the user selects the "Play" option from the screen shown in Figure 4, the MIDlet 9 reads the next random value from its store (step s101). Referring to Figure 7 also, the MIDlet 9 then generates an moving image 21 representing the spinning reels of an electro-mechanical slot machine (step s102). The "reels" are made to appear to stop in turn. The symbol (bell, bar, cherries, plum etc.) displayed on each "reel" when it stops is determined by the current random value.

[0043] Referring to Figure 8, when all of the "reels" have come to rest, a suitable commiseratory or congratulatory message 22 is added to the displayed image together with an invitation to play again 23 or return to the

main screen 24 (step s103).

[0044] At the end of the play, the MIDlet 9 determines whether the play was the last of purchased set of twenty plays (step s104). If the play was the last of a set, the MIDlet 9 sends a SOAP message to the origin server 1 which includes the ID of the completed play packet, the user's username and password, which may have been stored earlier or entered using the screen shown in Figure 4 (step s105) and the win values for each play. Assuming that the user is authenticated successfully, the request is passed to the second servlet 11b. The second servlet 11b retrieves the win amounts for the play set, identified by the play packet ID, from the database 8 and compares it with the win amounts received in the SOAP message. If the win amounts match, the second servlet 11b transfers any win amount to the user's account and sends an acknowledge response to the client 2, otherwise it sends an error response to the client.

[0045] If the user selects the play again option (step s106), flow returns to step s101, otherwise the MIDlet 9 terminates.

[0046] It will be appreciated that the user may be presented with other images, e.g. racing horses. Whatever the image or images represent, the user should be provided with the experience of playing a game of chance in real time.

[0047] In the foregoing, the user can win money amounts. However, this is not essential and the win values may have no monetary value. Also, the user may be billed for game packet by their mobile network operator. In such an embodiment, an additional step is built into the points registration process. The mobile device phone number or encrypted phone number is obtained via an http request from the mobile network operator, uploaded and registered in the gaming system database. This identifier is unique to each player, facilitates mobile network operator billing and reduces the information required at registration.

[0048] The above-described embodiment uses a mobile agent which communicates with an origin server via a mobile communication network using WAP and http. However, the present invention may be used to provide remote user interfaces connected to a central server by a wired network. In such a system, since the game is actually played on the server before the user interface displays the game playing image, the system provides improved security over conventional amusement arcade systems.

[0049] MIDlets have been employed in the above described embodiments because they are convenient for embodiments using mobile phones as the client. However, it will be appreciated that the client may be written using any suitable language and/or framework. Similarly, the server need not be based around an http server such as Apache or Microsoft IIS and the server may be custom built for putting the present invention into effect.

Claims

1. A method providing a remote user interface in a gaming system, the method comprising:

5 sending a play request message from a client to a remote server;
responding to the play request message at the server by generating a random play outcome value and transmitting the random play outcome value to the client; and
10 receiving said play outcome value at the client;

characterised by

15 receiving a game start user input at the client after said reception of the play outcome value; and
generating an image at the client in dependence on said play outcome value in response to reception of said game start input.

2. A method according to claim 1, wherein said play outcome value is a random value.

3. A method according to claim 1 or 2, wherein said image is a moving image and said play outcome value determines the final state of said image.

4. A method according to claim 1, 2 or 3, wherein a plurality of play outcome values are generated and transmitted to the client in response to one play request message therefrom.

5. A method according to any preceding claim, including the client reporting the generation of said image to the server.

6. A method according to any preceding claim, wherein communication between the client and the server uses http.

7. A gaming system comprising:

a client (2) including display means; and
a server (1) remote from the client;

wherein the client (2) is configured for sending a play request message to the remote server, receive a play outcome value, provided by the server in response to said play request message and the server (1) is configured for responding to the play request message by generating a random play outcome value and transmitting the random play outcome value to the client (2),

characterised by the client (2) being configured to respond to a user input, after receiving the play outcome value, to generate an image at the client in dependence on the received play outcome.

8. A system according to claim 7, wherein the play outcome value is a random value.
9. A system according to claim 7 or 8, wherein the client (2) is configured to generate a moving image and set the final state of the image in dependence on said play outcome value. 5
10. A system according to claim 7, 8 or 9, wherein the client (2) includes user input means (116). 10
11. A system according to any one of claims 7 to 10, wherein the server (1) is configured to generate and transmit a plurality of play outcome values in response to one play request message. 15
12. A system according to any one of claims 7 to 11, wherein the client (2) is configured for reporting the generation of said image to the server (1). 20
13. A server system for a gaming system having a remote user interface, the server system (1) being **characterised by** web server means (6) making a client program (9) available for download and dynamic content generating means (7), wherein the client program (9) is configured for sending a play request message from a client (2) to the web server means (6) and generating an image at the client (2) in dependence on a play outcome value, received from the web server means (6) in response to a play request message, and in response to a user input made after reception of said play outcome value, and the dynamic content generating means (7) is configured for responding to a play request message by generating a play outcome value and transmitting the random value to the requesting client (2). 25 30 35
14. A system according to claim 13, wherein the play outcome value is a random value. 40
15. A system according to claim 13 or 14, wherein the client program (9) is configured to generate a moving image at a client (2) and set the final state of the image in dependence on a play outcome value received in response to a play request message. 45
16. A system according to claim 13, 14 or 15, wherein the dynamic content generating means (7) is configured to generate and transmit a plurality of play outcome values in response to one play request message. 50
17. A system according to any one of claims 13 to 16, wherein the client program (9) is configured for reporting the generation of said image to the server (1). 55

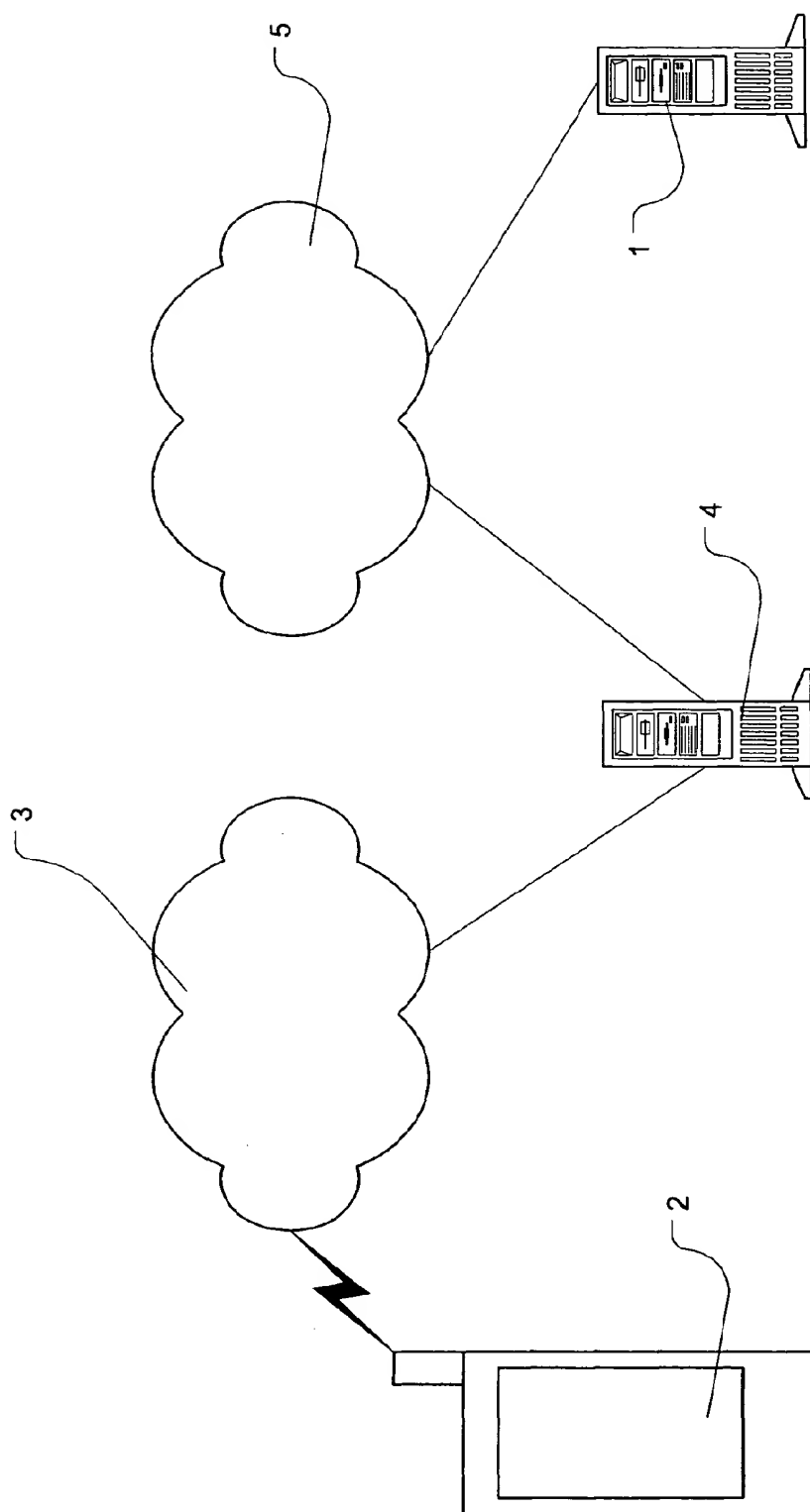


Figure 1

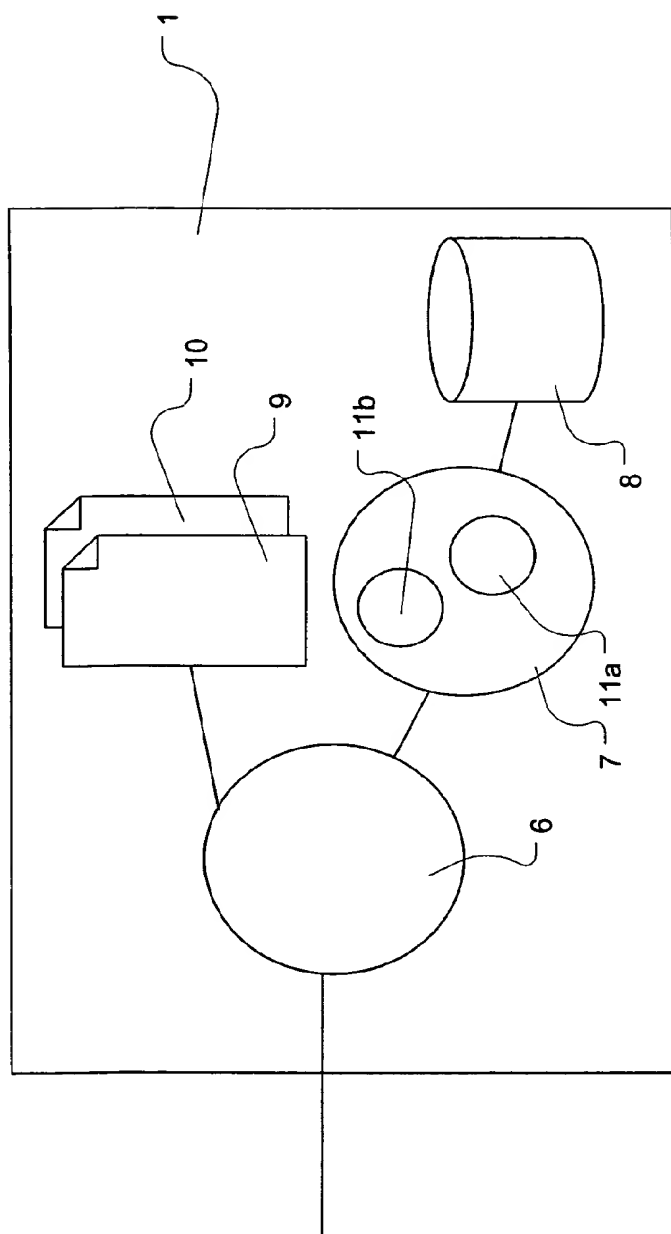


Figure 2

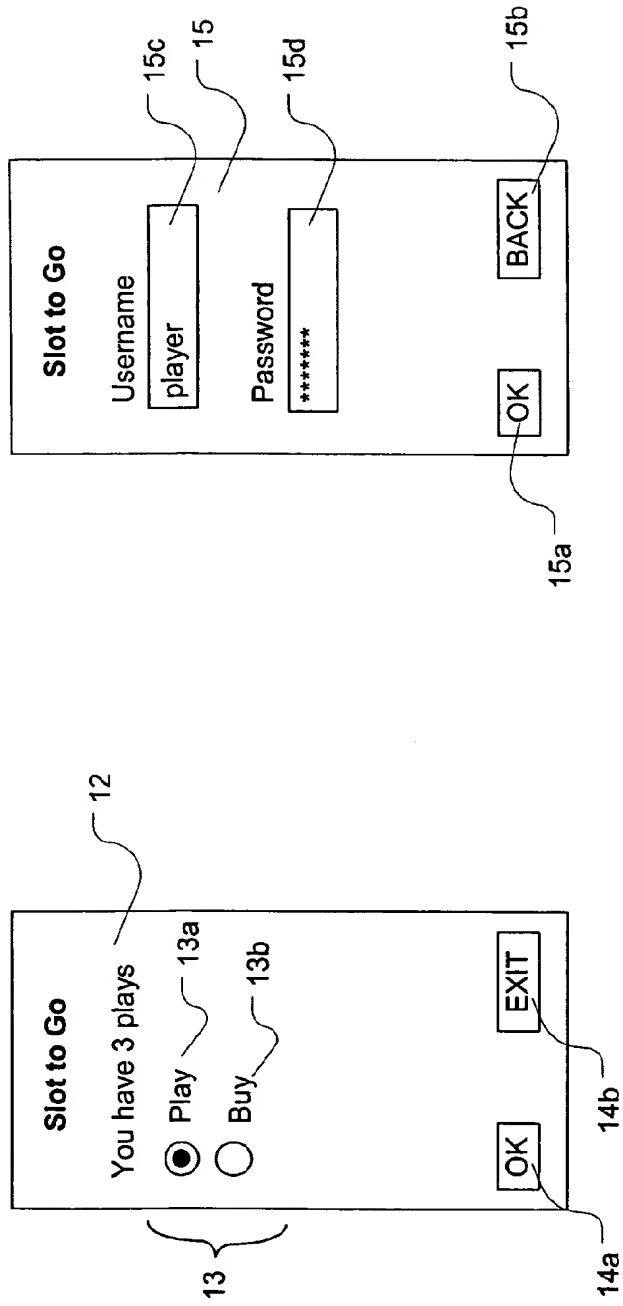


Figure 3

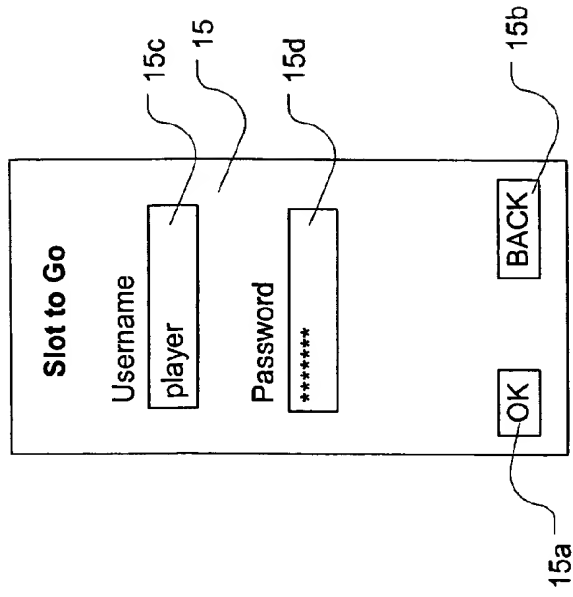


Figure 4

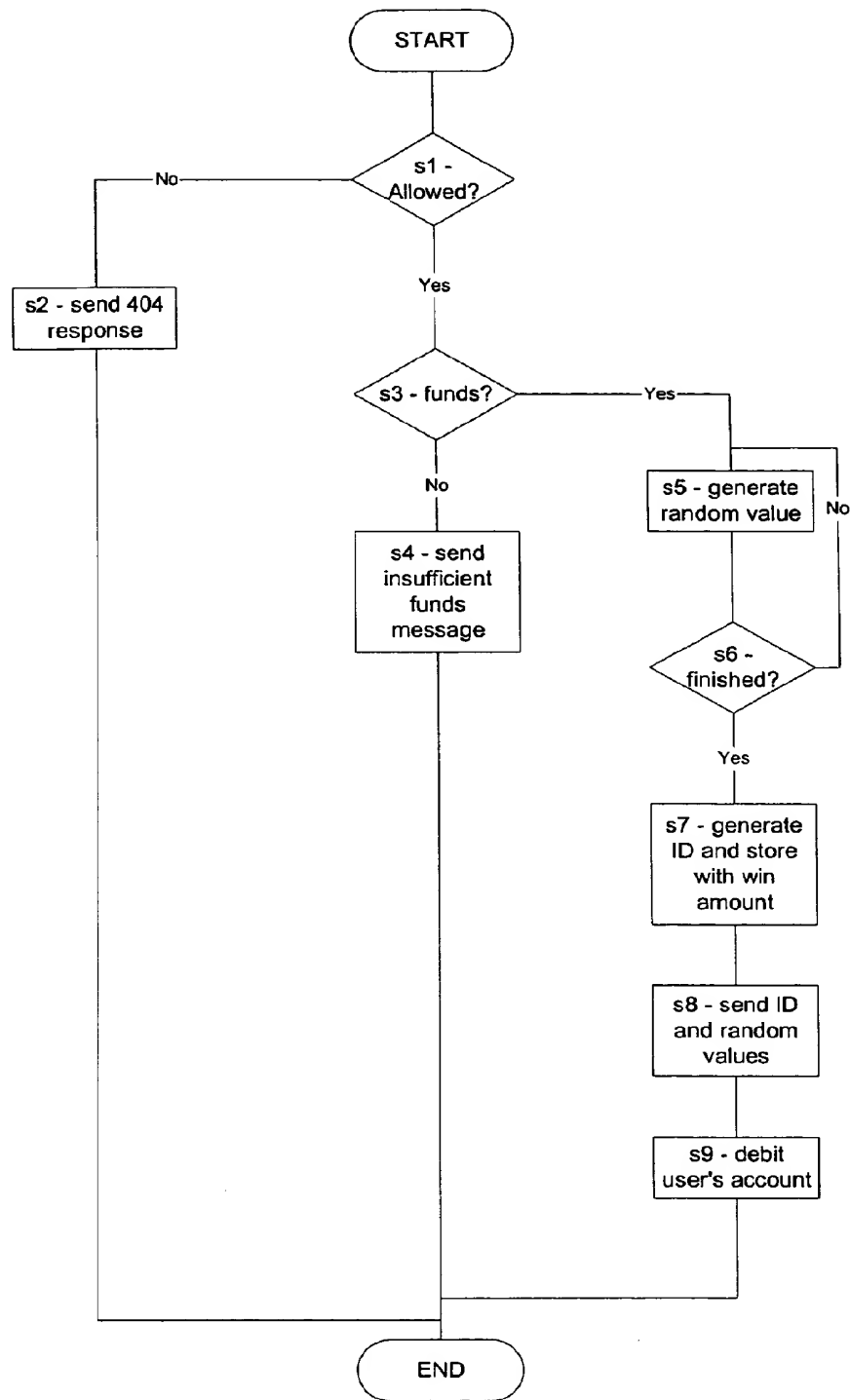


Figure 5

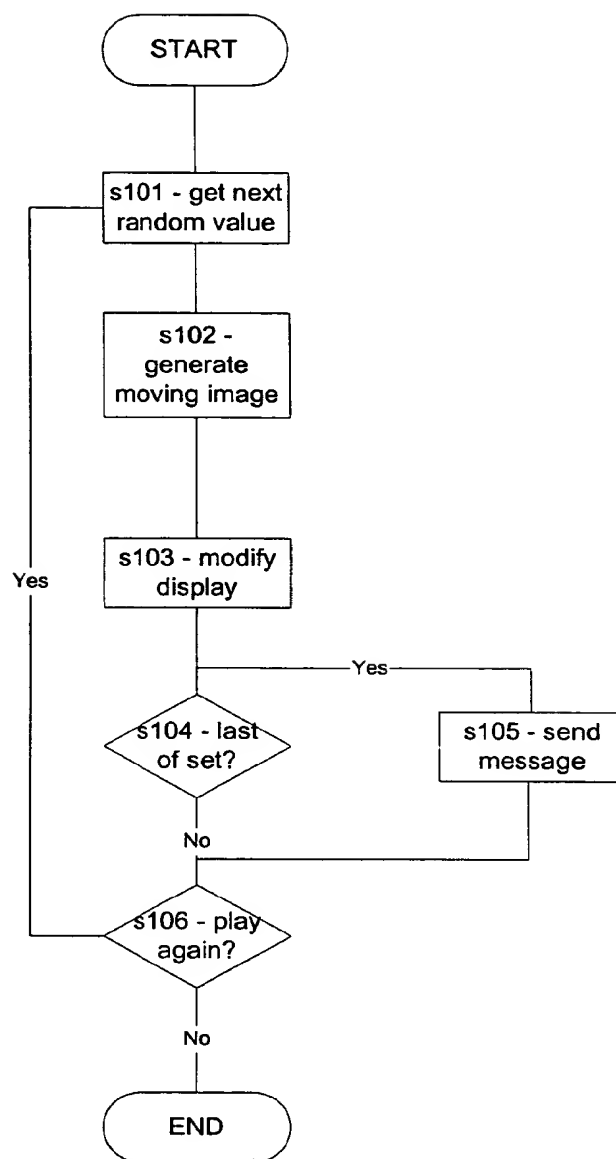


Figure 6

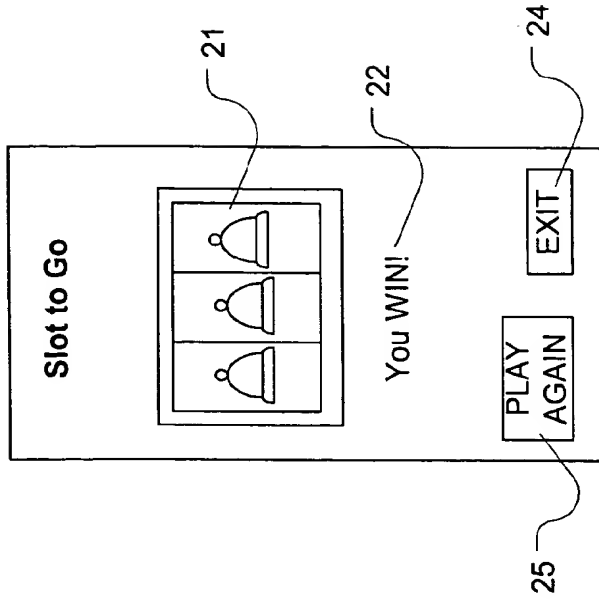


Figure 8

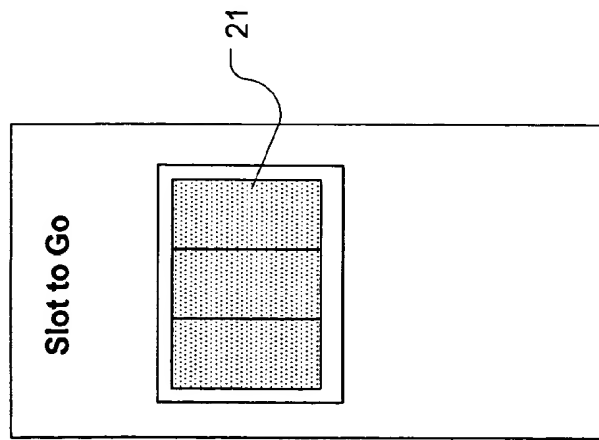


Figure 7